

ADMINISTRACIÓN DE SERVIDORES

---

## Memoria Práctica 1: Linux Básico

---



**Autores:**

Alberto Aragón Torralba  
Alejandro Guerrero Medina  
Gonzalo Sánchez Bazán

**Grado en Ingeniería Informática**  
Curso 2022 - 2023

## Índice

<b>1. Preparación del entorno</b>	<b>2</b>
1.1. Ejercicio 1 . . . . .	2
<b>2. Órdenes básicas</b>	<b>3</b>
2.1. Ejercicio 2 . . . . .	3
2.2. Ejercicio 3 . . . . .	3
2.3. Ejercicio 4 . . . . .	3
2.4. Ejercicio 5 . . . . .	4
<b>3. Procesos</b>	<b>5</b>
3.1. Ejercicio 6 . . . . .	5

# 1. Preparación del entorno

## 1.1. Ejercicio 1

Primero debemos crear una máquina Ubuntu con **Vagrant**. En este documento usaremos como referencia la *box*: **ubuntu/bionic64**, pero se puede usar cualquier otra que sea, al menos, una máquina Ubuntu.

En una terminal de comandos, deberemos ejecutar el siguiente comando:

```
vagrant init ubuntu/bionic64
```

Con lo que **Vagrant** inicializará una máquina con la *box* indicada, junto con su respectivo *Vagrantfile*, donde configuraremos los siguientes aspectos:

- La configuración red indicada en el enunciado, donde el puerto **80** de la máquina invitada es redireccionado al puerto **8080** del anfitrión.
- *Apache* instalado en la máquina, donde el provisioner *shell* ejecuta la configuración inicial de la máquina.

En esencia, el fichero *Vagrantfile*, obviando la configuración por defecto, debería de contener las siguientes instrucciones añadidas o modificadas:

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/bionic64"
  config.vm.network "forwarded port", guest: 80, host: 8080
  config.vm.provision "shell", inline: <<-SHELL
    apt-get update
    apt-get install -y apache2
  SHELL
end
```

Una vez preparada la máquina con su respectiva configuración, ejecutando las líneas de comando:

```
vagrant up
vagrant ssh
```

Creemos la máquina acorde a la configuración del *Vagrantfile* asociado y nos conectaremos por *ssh* a la máquina invitada, ya preparada para poder trabajar en ella.

## 2. Órdenes básicas

### 2.1. Ejercicio 2

ENUNCIADO: Muestre todos los ficheros de C (.c, .h) del sistema.

SOLUCIÓN:

```
sudo find / -type f \( -name "*.c" -o -name "*.h" \)
```

EXPLICACIÓN: El comando `find` resulta ser ideal en casos donde se requiera hacer una búsqueda dentro del sistema para localizar archivos de un tipo en directorios asociados, tal y como ocurre en el enunciado especificado. En su semántica, encontramos:

- `sudo`: Para evitar mostrar avisos referidos a *permiso denegado a fichero*.
- `type -f`: Indicamos que el tipo de archivo que estamos buscando es de tipo **común**.
- `\( -name "*.c" -o -name "*.h" \)`: Expresión regular en la que indicamos mediante las *flags* `-name` todo archivo que sea `.c` o `.h`.

### 2.2. Ejercicio 3

ENUNCIADO: Modifique la solución del ejercicio anterior para que no se muestren los errores.

SOLUCIÓN:

```
sudo find / -type f \( -name "*.c" -o -name "*.h" \) 2> /dev/null
```

EXPLICACIÓN: Para este ejercicio, lo único que tendremos que añadir a la sentencia anterior será la expresión `2>`, lo que hace redireccionar el flujo estándar de errores (`stderr`) al fichero indicado; `/dev/null`, un fichero especial cuyo propósito es descartar toda información contenida.

### 2.3. Ejercicio 4

ENUNCIADO: Muestre por pantalla cuántos usuarios no pueden iniciar sesión.

SOLUCIÓN:

```
grep -F 'nologin' /etc/passwd | cut -d: -f1
```

EXPLICACIÓN: El comando `grep` busca un patrón definido en un archivo de texto. Es una de las herramientas más versátiles y útiles que ofrece *Unix* junto con el comando `find`. Veamos con más detalle el modo de empleo en este enunciado:

- `grep -F 'nologin' /etc/passwd`: Especificamos que se interpreten los patrones como cadenas literales, en este caso `nologin`, en el fichero `/etc/passwd`.
- `cut -d: -f1`: El comando `cut` elimina secciones del fichero especificado que se sigue de la *pipe*. Al tratarse de un fichero cuya semántica corresponde a la de un fichero *CSV*, hemos de indicar el delimitador mediante la *flag* `-d:`, y por último seleccionamos con la siguiente *flag* `-f1` el campo que nos interesa del fichero `/etc/passwd`, siendo este el primero pues corresponde al nombre del usuario.

SOLUCIÓN ALTERNATIVA:

```
sudo passwd -S -a | awk '$2 ~ /L/ {print $1}'
```

EXPLICACIÓN:

- `sudo passwd -S -a`: Muestra la información sobre todos los usuarios en el sistema, incluyendo su estado de contraseña, gracias a las *flags* `-a` y `-S`, respectivamente.
- `awk '$2 ~ /L/ {print $1}'`: Si la segunda columna es “L” significa que la cuenta está bloqueada y no puede iniciar sesión. En la primera columna se muestra el nombre de usuario.

## 2.4. Ejercicio 5

**ENUNCIADO:** Muestre los distintos tipos de inicio de sesión de los usuarios que están dados de alta en el sistema, así como el número de usuarios que tienen cada tipo. Ordene dicha información por el número de usuarios.

SOLUCIÓN:

```
cut -d: -f7 /etc/passwd | sort | uniq -c | sort -nr
```

EXPLICACIÓN: Dado que ya se ha explicado el funcionamiento del comando `cut`, se procederá con la explicación de los nuevos comandos: `sort` y `uniq`. El primero ordena los resultados del fichero especificado según que *flags* se indiquen y el segundo omite aquellos resultados repetidos por pantalla. Siguiendo la naturaleza del ejercicio pedido, tendríamos:

- `cut -d: -f7 /etc/passwd | sort`: Seleccionamos el séptimo campo del fichero `/etc/passwd`, cuya correspondencia es el tipo de inicio de sesión del que dispone el usuario (recordemos que se trata de un fichero *CSV*, de ahí el delimitador `-d:`), y ordenamos la búsqueda.
- `uniq -c`: La *flag* `-c` añade por pantalla el número de ocurrencias repetidas del registro.
- `sort -nr`: Finalmente, ordenamos los tipos de inicio de sesión por número de usuarios (es decir, número de ocurrencias) y en orden descendente, gracias a las *flags* `-nr`, respectivamente.

SOLUCIÓN ALTERNATIVA:

```
cat /etc/passwd | awk -F: '{print $NF}' | sort | uniq -c | sort -nr
```

EXPLICACIÓN: Esta alternativa plantea otro medio para obtener la información gracias al uso del comando `cat`, seguido del comando `awk`.

- `cat /etc/passwd | awk -F: '{print $NF}'`: Copia el contenido del fichero `/etc/passwd` y lo muestra por pantalla mediante `awk`, un lenguaje de programación especializado en la manipulación textual de datos. Con la *flag* `-F` definimos el separador `:` e invocamos seguidamente la función `print` bajo la variable especial `NF`, cuyo propósito es mostrar el número de campos de la instancia actual.
- El resto de comandos en las siguientes *pipes* son idénticos a la propuesta anterior.

## 3. Procesos

### 3.1. Ejercicio 6

ENUNCIADO: Encuentre los programas con el setgid activado y guarde el resultado en *programassetgid.txt*.

SOLUCIÓN:

```
sudo find / -perm /g=s -executable > programassetgid.txt
```

EXPLICACIÓN: Dado que ya se ha explicado el funcionamiento del comando `find`, se procederá con la explicación de las *flags* utilizadas:

- `-perm /g=s -executable`: Buscamos archivos con los permisos especificados `/g=s`, argumento que establece que los archivos buscados han de tener permisos establecidos para el grupo, y que sean ejecutables (`-executable`).
- `> programassetgid.txt`: Redireccionamos la salida de la búsqueda realizada al archivo de texto especificado, guardándolo en el directorio actual.